(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 081 868 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
07.03.2001 Bulletin 2001/10

(51) Int. Cl.[7]: **H03M 13/41**, G11B 20/10, H04L 25/497

(21) Application number: **00118373.0**

(22) Date of filing: **24.08.2000**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **03.09.1999 US 152477 P**
14.02.2000 US 503534

(71) Applicant:
Infineon Technologies North America Corp.
San Jose, CA 95112-6000 (US)

(72) Inventors:
- **Ashley, Jonathan**
  Los Gatos, California 95030 (US)
- **Karabed, Razmik**
  San Jose, California 95120 (US)

(74) Representative:
Patentanwälte
Westphal, Mussgnug & Partner
Waldstrasse 33
78048 Villingen-Schwenningen (DE)

(54) Supporting an E2PRML-type channel and an EPRML-type channel with the same trellis structure

(57) A Viterbi trellis is provided which allows for implementation of either an EPRML type channel or an E2PRML type channel (208) using a single trellis structure. According to a specific embodiment, a trellis is provided which can support a 0 mod 2 EPRML (M2EPRML) channel and a modified E2PRML (ME2PRML) channel.

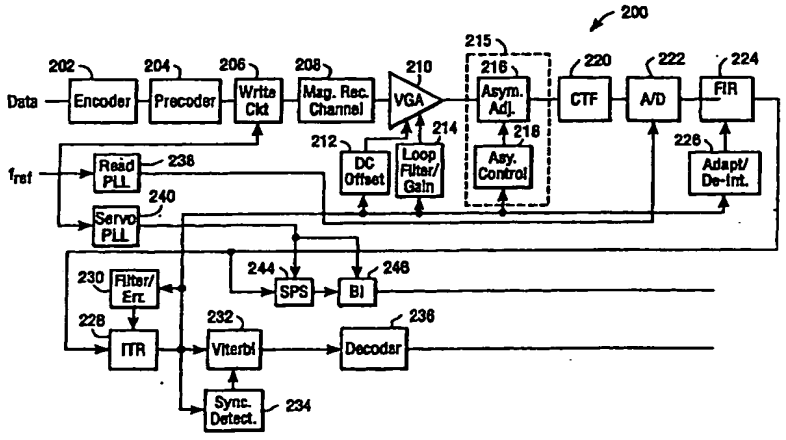

FIG. 1

EP 1 081 868 A2

Description

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority from U.S. Provisional Application Serial No. 60/152,477, filed September 3, 1999.

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0002] The present invention relates to encoding for disk drives and, particularly, to an improved Viterbi trellis for sampled amplitude read channels.

DESCRIPTION OF THE RELATED ART

[0003] In order to achieve higher recording densities and reduce intersymbol interference, designers of magnetic recording channels have switched from analog peak detection techniques to sampled data detection techniques. In sampled data detection systems, the readback signal is filtered and sampled at a channel rate of 1/T, where T is the duration of a channel symbol.

[0004] One such technique is referred to as "partial response with maximum likelihood" (PRML). In PRML systems, the output of the noisy partial response channel is sampled at the channel rate and detected using a maximum likelihood Viterbi detector.

[0005] The partial response channel has a transfer function of the form $(1-D)(1+D)$ or $1-D^2$, where D represents a unit time delay operator with unit-time T. Thus, the noiseless output of the partial response channel is equal to the input signal minus a version of the input delayed in time by period 2T.

[0006] To further increase recording density and decrease the need for equalization, higher order PRML systems have been developed. The extended partial response with maximum likelihood (EPRML) channel has a transfer function of the form $(1-D)(1+D)^2$ or $(1+D-D^2-D^3)$. Thus, the noiseless output of the extended partial response channel is equal to the input signal minus a version of the input signal delayed in time by 2T, minus a version of the input signal delayed in time by 3T and plus a version of the input signal delayed in time by T. Similarly, the E2PRML channel has a transfer function of the form $(1-D)(1+D)^3$.

[0007] As noted above, Viterbi decoders are typically employed in sampled amplitude channels. Viterbi decoders are specific implementations of the Viterbi algorithm. A Viterbi detector unit is based on periodic examination of metrics associated with alternate sequences of recorded bits, wherein each sequence is typically labeled as a "path" and the associated metric is designated a "path metric." The most probable correct path is then determined by choosing a minimum path metric based on an iterative process involving successive comparison of associated path metrics.

[0008] In particular, two paths within a constrained, predetermined path length are examined. Since the recorded bit only depends on the constraint length corresponding to a finite number of neighbor bits, it becomes possible to abandon the path associated with the larger of the two path metrics corresponding to each path pair. Consequently, the number of possible paths can be restricted to a finite value by abandoning all but one of the total number of paths each time a new bit is added and examined during the data detection procedure. This process of path abandonment in order to compute the best path to each node of the trellis is executed by a sequence of operations commonly referred to as add-compare-select or ACS.

[0009] In many instances of implementing sampled amplitude channels, for example, it is desirable to implement a system of more than one partial response channel type. Conventionally, this has required implementation of unique trellises for each type. This is disadvantageous in that implementation costs are relatively increased. Further, implementing multiple partial response channel types each with its own trellis on a single integrated circuit chip can require different amounts of chip space, which can result in sub-optimal chip space usage.

SUMMARY OF THE INVENTION

[0010] These and other drawbacks in the prior art are overcome in large part by a system and method according to the present invention. In particular, the present invention allows for implementation of either an EPRML type channel or an E2PRML type channel using a single trellis structure. According to one implementation, a trellis is provided which can support a 0 mod 2 (i.e., there are an even number of 1's in every codeword) EPRML (M2EPRML) channel and a modified E2PRML (ME2PRML) channel.

[0011] According to one implementation, a sixteen-state trellis is provided. Sixteen of thirty-two edges in the

M2EPRML carry the same non-return-to-zero (NRZ) values as the corresponding edges of the ME2PRML trellis, and sixteen of thirty-two edges carry the same sample values. The trellis supports M2EPRML if sixteen of its edges are eliminated every P clock cycles, where P is the block length of the code. The trellis can support ME2PRML sequences if four of its edges are eliminated every k mod 9 (k in {1, 2, 4, 6, 8}) clock cycles.

[0012] A better understanding of the invention is obtained when the following detailed description is considered in conjunction with the following drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013]

FIG. 1 is a diagram of an exemplary read/write channel according to the present invention;
FIG. 2 is a diagram illustrating an exemplary trellis implementation for the read/write channel of FIG. 1;
FIG. 3 is a diagram of a pair of trellises having identical connectivity;
FIG. 4 is a diagram of common NRZ values for the trellises of FIG. 2;
FIG. 5 is a diagram of common sample values for the trellises of FIG. 2;
FIG. 6 is a diagram of M2EPRML and ME2PRML implementation using the trellises of FIG. 3; and
FIG. 7 is a state transition diagram for states of the trellises of FIG. 4.

DETAILED DESCRIPTION OF THE INVENTION

[0014] FIGs. 1-7 illustrate a trellis implementation according to the present invention. Briefly, the trellis described employs the same structure to implement Viterbi detectors for an EPRML type channel and an E2PRML type channel.

[0015] Turning now to the drawings and, with particular attention to FIG. 1, a block diagram of a sampled amplitude read channel according to an embodiment of the invention is shown and identified by the reference numeral 200. As will be discussed in greater detail below, the sampled amplitude read channel 200 may implement an E2PRML type channel or an EPRML type channel.

[0016] During a write operation, data is written onto the media. The data is encoded in an encoder 202, such as a run length limited (RLL) or other encoder. A precoder 204 precodes the sequence to compensate for the transfer function of the magnetic recording channel 208 and equalizing filters. As will be discussed in greater detail below, the precoder/encoder may encode the data such that after precoding the data has a pre-determined parity structure. Further details on such decoding may be obtained from U.S. Patent 5,809,081, which is hereby incorporated by reference in its entirety as if fully set forth herein. Turning back to FIG. 1, the write circuitry 206 modulates the current in the recording head coil to record a binary sequence onto the medium. A reference frequency f_{ref} provides a write clock to the write circuitry 206.

[0017] The bit sequence is then provided to a variable gain amplifier 210 to adjust the amplitude of the signal. DC offset control 212 and loop filter/gain error correction 214 may be provided to control the adjustment of the VGA 210. Further, an asymmetry control unit 215 including an asymmetry adjustment unit 216 and asymmetry control 218 may be provided to compensate for magneto-resistive asymmetry effects.

[0018] The signal is then provided to a continuous time filter 220, which may be a Butterworth filter, for example, to attenuate high frequency noise and minimize aliasing into baseband after sampling. The signal is then provided to an analog to digital converter 222 to sample the output of the continuous time filter 220.

[0019] A finite impulse response filter 224 provides additional equalization of the signal to the desired response. The output of the FIR 224 is provided to an interpolated timing recovery unit 228, which is used to recover the discrete time sequence. The output of the interpolated timing recovery unit is used to provide a feedback control to the DC offset control 212, the gain error 214, the asymmetry control 218 and the FIR 224 control 226. The output of the interpolated timing recovery 228 is provided to a Viterbi detector 232 to provide maximum likelihood detection. Further, the ITR output is provided to a sync detector 234. Sync mark information is then provided to the Viterbi detector 232 for use in sequence detection. The Viterbi detector output is then provided to the decoder 236 which decodes the encoding provided by the encoder 202. The Viterbi detector 232 may implement either of the trellises described below.

[0020] FIG. 2 illustrates a Viterbi decoder employing a trellis according to the present invention. The Viterbi decoder includes a branch metric generator 100, an add-carry-select (ACS) unit 102, and a survivor memory 104. The branch metric generator 100 receives coded data and calculates a branch metric, which is a distance between codes on each branch corresponding to a received signal. As will be discussed in greater detail below, in implementing a nonstationary EPRML type channel, a predetermined number of paths between nodes are constrained to be not valid In particular, in one implementation, when a sample corresponding to a last coordinate of a codeword reaches the detector every Pth clock (where P is the block length of the code), the edges are not valid. Similarly, in implementing an E2PRML type channel, four edges are eliminated every k mod 9 clock cycles, where k = {1, 2, 4, 6, 8}.

[0021] The ACS unit 102 receives the branch metrics from the branch metric generator 100, adds them to the previous path metric, and determines a plurality of candidate paths. The ACS 102 then compares the plurality of ACS path metric values and selects a path having the shortest path metric, and outputs the newly selected path metric and the compared result, namely the decision bit. The ACS unit 102 updates the path metric by using the branch metric obtained from the branch metric generator 100 at each decoding cycle and outputs a decision bit for every state in the trellis. The survivor memory 104 stores the decision bit obtained from the ACS unit 102, receives the original information sequence by using the decision bit, and outputs it as Viterbi-decoded data.

[0022] The trellises are illustrated in greater detail with reference to FIG. 3. As shown, a pair of trellises, T1 and T2, having identical connectivity are illustrated. In FIGs. 3-6, it is noted that a solid arrow means the NRZ state is a "1" and a dotted arrow means the NRZ state is a "0".

[0023] The following provides the general framework for the implementation described below:

[0024] Let Finite State Transition Diagram (FSTD) H1 be defined as follows:

States of H1: They are denoted by binary n-tuples (a1 a2 a3 ... an), where ai = 0 or 1,
Edges of H1: There is an edge, e, from (a1 a2 a3 ... an) to (b1 b2 b3 ... bn) if and only if (b1 b2 b3 ... b(n-1))=(a2 a3 ... an). Edge e is labeled bn.

[0025] Let FSTD H2 be defined as:

States of H2: They are denoted by binary n-tuples (p u(n-1) u(n-2) u(n-3) ... u1), where ui = 0 or 1 and p = 0 or 1,
Edges of H2: There is an edge, e, from (p1 u(n-1) u(n-2) u(n-3) ... u1) to (p2 v(n-1) v(n-2) v(n-3) ... v1) if and only if (v(n-2) v(n-3) ... v1)=(u(n-1) u(n-2) u(n-3) ...u2), and p2=v(n-1) ⊕ p1. (⊕ denotes exclusive-OR operation). Edge e is labeled v(n-1).
Theorem 1: The di-graphs H1 and H2 are isomorphic.
Proof: Let map F be from states of H1 to states of H2, where F((a1 a2 a3 ... an)) = (an, a(n-1) ⊕ an, ... a4 ⊕ a3, a3 ⊕ a2, a2 ⊕ a1).

[0026] Map F is one-to-one; moreover, it establishes the isomorphism.

[0027] The trellis of H1 can support PRML (or an extended PRML) constraint, and the trellis of H2 can support PRML with parity (or an extended PRML with parity) constraint.

[0028] Using Theorem 1 we can prove the following.

Theorem 2: If h1(D) and h2(D) are two channel filters with orders n and n-1, respectively, then the FSTD supporting h1(D) is isomorphic to the FSTD supporting h2(D) with parity.
Proof: Let H1 and H2 support h1(D) and h2(D) sequences, respectively, and let the coordinate p denote the parity information.

[0029] Thus if h1(D) and h2(D) are two channel filters with orders n and n-1, respectively, then one can use the same underlying trellis to detect both h1(D) and h2(D) with parity.

[0030] Such an implementation is described below. The trellises T1 and T2 of FIG. 3 illustrate NRZ (non-return-to-zero) states. Using the NRZ convention, a one (1) represents a particular direction of magnetization, and a zero (0) another direction of magnetization. Each trellis T1, T2 includes sixteen (16) states (states 0 to 16, represented in binary as 0000 to 1111). Each of the sixteen states represents four consecutive magnetization states. The notation X/Y represents X = NRZ values and Y = sample values. As is known, EPR sequences take on sample values [0,1,2,1,0] for an isolated pulse, and E2PR sequences take on the sample values [0,1,3,3,1,0] for an isolated pulse.

[0031] As can be seen more clearly with reference to FIG. 4, sixteen of thirty-two edges in T1 carry the same NRZ values as the corresponding edges of trellis T2. Further, as shown in FIG. 5, sixteen of thirty-two edges in T1 carry the same sample values as their corresponding edges of T2 (though do not necessarily have the same NRZ values). Thus, a Viterbi detector based on trellis T1 can be modified to operate on trellis T2 by replacing 16 sample values and 16 NRZ values of T1.

[0032] In particular, as shown in FIG. 6, T1 can support an EPRML type system if sixteen (16) of its edges are "eliminated" every P clock cycles, where P is the block length of the code. In particular, the edges 602-632 are eliminated. These edges are "eliminated" every time a sample that corresponds to a last coordinate of a codeword reaches the Viterbi detector. In other words, for such a sample, the edges are not valid. In one implementation, P = 51.

[0033] Similarly, as shown in FIG. 6, the trellis T2 can support ME2PRML sequences if four (4) of its edges are "eliminated" every clock = k mod 9, for k in (1, 2, 4, 6, 8). In particular, the edges 650-656 are eliminated.

[0034] State transitions for the trellises T1 and T2 are defined more particularly with reference to FIG. 7 and Table 1 and 2 below. In particular, FIG. 7 illustrates three states: an end state M, and two beginning states K and L. As can

be seen, the transitions from states K and L are associated with NRZ labels u1, u2, respectively, and sample labels, v1 and v2, respectively. Thus, transitions to state M may be defined as follows:

```
Previous_state_1[M]=K
Previous_state_2[M]=L
Edge_label_nrz_1[M] = u1
Edge_label_nrz_2[M] = u2
Edge_label_sample_1[M]=v1
Edge_label_sample_2[M]=v2
```

[0035] The T1 trellis is defined with reference to Table 1 below:

```
number_of_states=18;

previous_state_1[0]=0 ;    previous_state_2[0]=1;
previous_state_1[1]=2 ;    previous_state_2[1]=3 ;
previous_state_1[2]=4 ;    previous_state_2[2]=5 ;
previous_state_1[3]=16 ;   previous_state_2[3]=7 ;
previous_state_1[4]=8 ;    previous_state_2[4]=9 ;
previous_state_1[5]=10 ;   previous_state_2[5]=11 ;
previous_state_1[6]=12 ;   previous_state_2[6]=13 ;
previous_state_1[16]=12 ;  previous_state_2[16]=13 ;
previous_state_1[7]=14 ;   previous_state_2[7]=15 ;
previous_state_1[8]=8 ;    previous_state_2[8]=9 ;
previous_state_1[9]=10 ;   previous_state_2[9]=11 ;
previous_state_1[10]=12 ;  previous_state_2[10]=13 ;
previous_state_1[11]=17 ;  previous_state_2[11]=15 ;
previous_state_1[12]=0 ;   previous_state_2[12]=1 ;
previous_state_1[13]=2 ;   previous_state_2[13]=3 ;
previous_state_1[14]=4 ;   previous_state_2[14]=5 ;
previous_state_1[17]=4 ;   previous_state_2[17]=5 ;
previous_state_1[15]=6 ;   previous_state_2[15]=7 ;


edge_label_nrz_1[0]=0 ; edge_label_sample_1[0]=0 ;
edge_label_nrz_2[0]=0 ; edge_label_sample_2[0]=-1 ;
edge_label_nrz_1[1]=0 ; edge_label_sample_1[1]=-1 ;
edge_label_nrz_2[1]=0 ; edge_label_sample_2[1]=-2 ;
edge_label_nrz_1[2]=0 ; edge_label_sample_1[2]=1 ;
edge_label_nrz_2[2]=0 ; edge_label_sample_2[2]=0 ;
edge_label_nrz_1[3]=0 ; edge_label_sample_1[3]=0 ;
edge_label_nrz_2[3]=0 ; edge_label_sample_2[3]=-1 ;
edge_label_nrz_1[4]=1 ; edge_label_sample_1[4]=1 ;
edge_label_nrz_2[4]=1 ; edge_label_sample_2[4]=0 ;
edge_label_nrz_1[5]=1 ; edge_label_sample_1[5]=0 ;
edge_label_nrz_2[5]=1 ; edge_label_sample_2[5]=-1 ;
edge_label_nrz_1[6]=1 ; edge_label_sample_1[6]=2 ;
edge_label_nrz_2[6]=1 ; edge_label_sample_2[6]=1 ;
edge_label_nrz_1[16]=1 ; edge_label_sample_1[16]=2 ;
edge_label_nrz_2[16]=1 ; edge_label_sample_2[16]=1 ;
edge_label_nrz_1[7]=1 ; edge_label_sample_1[7]=1 ;
edge_label_nrz_2[7]=1 ; edge_label_sample_2[7]=0 ;
```

edge_label_nrz_1[8]=0 ; edge_label_sample_1[8]=0 ;
edge_label_nrz_2[8]=0 ; edge_label_sample_2[8]=-1 ;
edge_label_nrz_1[9]=0 ; edge_label_sample_1[9]=-1 ;
edge_label_nrz_2[9]=0 ; edge_label_sample_2[9]=-2 ;
edge_label_nrz_1[10]=0 ; edge_label_sample_1[10]=1 ;
edge_label_nrz_2[10]=0 ; edge_label_sample_2[10]=0 ;
edge_label_nrz_1[11]=0 ; edge_label_sample_1[11]=0 ;
edge_label_nrz_2[11]=0 ; edge_label_sample_2[11]=-1 ;
edge_label_nrz_1[12]=1 ; edge_label_sample_1[12]=1 ;
edge_label_nrz_2[12]=1 ; edge_label_sample_2[12]=0 ;
edge_label_nrz_1[13]=1 ; edge_label_sample_1[13]=0 ;
edge_label_nrz_2[13]=1 ; edge_label_sample_2[13]=-1 ;
edge_label_nrz_1[14]=1 ; edge_label_sample_1[14]=2 ;
edge_label_nrz_2[14]=1 ; edge_label_sample_2[14]=1 ;
edge_label_nrz_1[17]=1 ; edge_label_sample_1[17]=2 ;
edge_label_nrz_2[17]=1 ; edge_label_sample_2[17]=1 ;
edge_label_nrz_1[15]=1 ; edge_label_sample_1[15]=1 ;
edge_label_nrz_2[15]=1 ; edge_label_sample_2[15]=0 ;

Table 1

[0036] The T2 trellis is described with reference to Table 2 below:

```
number_of_states=18;

previous_state_1[0]=0 ;    previous_state_2[0]=8;
previous_state_1[1]=0 ;    previous_state_2[1]=8 ;
previous_state_1[2]=1 ;    previous_state_2[2]=9 ;
previous_state_1[16]=1 ;   previous_state_2[16]=9 ;
previous_state_1[3]=1 ;    previous_state_2[3]=9 ;
previous_state_1[4]=16 ;   previous_state_2[4]=10 ;
previous_state_1[5]=2 ;    previous_state_2[5]=10 ;
previous_state_1[6]=3 ;    previous_state_2[6]=11 ;
previous_state_1[7]=3 ;    previous_state_2[7]=11 ;
previous_state_1[8]=4 ;    previous_state_2[8]=12 ;
previous_state_1[9]=4 ;    previous_state_2[9]=12 ;
previous_state_1[10]=5 ;   previous_state_2[10]=13 ;
previous_state_1[11]=5 ;   previous_state_2[11]=17 ;
previous_state_1[12]=6 ;   previous_state_2[12]=14 ;
previous_state_1[13]=6 ;   previous_state_2[13]=14 ;
previous_state_1[17]=6 ;   previous_state_2[17]=14 ;
previous_state_1[14]=7 ;   previous_state_2[14]=15 ;
previous_state_1[15]=7 ;   previous_state_2[15]=15 ;

edge_label_nrz_1[0]=0 ; edge_label_sample_1[0]=0 ;
edge_label_nrz_2[0]=0 ; edge_label_sample_2[0]=-1 ;
edge_label_nrz_1[1]=1 ; edge_label_sample_1[1]=1 ;
edge_label_nrz_2[1]=1 ; edge_label_sample_2[1]=0 ;
```

edge_label_nrz_1[2]=0 ; edge_label_sample_1[2]=2 ;
edge_label_nrz_2[2]=0 ; edge_label_sample_2[2]=1 ;
edge_label_nrz_1[16]=0 ; edge_label_sample_1[16]=2 ;
edge_label_nrz_2[16]=0 ; edge_label_sample_2[16]=1 ;
edge_label_nrz_1[3]=1 ; edge_label_sample_1[3]=3 ;
edge_label_nrz_2[3]=1 ; edge_label_sample_2[3]=2 ;
edge_label_nrz_1[4]=0 ; edge_label_sample_1[4]=0 ;
edge_label_nrz_2[4]=0 ; edge_label_sample_2[4]=-1 ;
edge_label_nrz_1[5]=1 ; edge_label_sample_1[5]=1 ;
edge_label_nrz_2[5]=1 ; edge_label_sample_2[5]=0 ;
edge_label_nrz_1[6]=0 ; edge_label_sample_1[6]=2 ;
edge_label_nrz_2[6]=0 ; edge_label_sample_2[6]=1 ;
edge_label_nrz_1[7]=1 ; edge_label_sample_1[7]=3 ;
edge_label_nrz_2[7]=1 ; edge_label_sample_2[7]=2 ;
edge_label_nrz_1[8]=0 ; edge_label_sample_1[8]=-2 ;
edge_label_nrz_2[8]=0 ; edge_label_sample_2[8]=-3 ;
edge_label_nrz_1[9]=1 ; edge_label_sample_1[9]=-1 ;
edge_label_nrz_2[9]=1 ; edge_label_sample_2[9]=-2 ;
edge_label_nrz_1[10]=0 ; edge_label_sample_1[10]=0 ;
edge_label_nrz_2[10]=0 ; edge_label_sample_2[10]=-1 ;
edge_label_nrz_1[11]=1 ; edge_label_sample_1[11]=1 ;
edge_label_nrz_2[11]=1 ; edge_label_sample_2[11]=0 ;
edge_label_nrz_1[12]=0 ; edge_label_sample_1[12]=-2 ;
edge_label_nrz_2[12]=0 ; edge_label_sample_2[12]=-3 ;
edge_label_nrz_1[13]=1 ; edge_label_sample_1[13]=-1 ;
edge_label_nrz_2[13]=1 ; edge_label_sample_2[13]=-2 ;
edge_label_nrz_1[17]=1 ; edge_label_sample_1[17]=-1 ;
edge_label_nrz_2[17]=1 ; edge_label_sample_2[17]=-2 ;
edge_label_nrz_1[14]=0 ; edge_label_sample_1[14]=0 ;
edge_label_nrz_2[14]=0 ; edge_label_sample_2[14]=-1 ;
edge_label_nrz_1[15]=1 ; edge_label_sample_1[15]=1 ;
edge_label_nrz_2[15]=1 ; edge_label_sample_2[15]=0 ;

Table 2

[0037] The mapping between the T1 and T2 states is shown in Table 3 below:

Table 3

ME2PRML States	M2EPRML states
0	0
1	12
2	6
3	10
4	3
5	15

Table 3 (continued)

ME2PRML States	M2EPRML states
6	5
7	9
8	1
9	13
10	7
11	11
12	2
13	14
14	4
15	8
16	16
17	17

[0038] It is noted that, while shown above implemented with 18 states, the above trellis may be implemented with 16 states as shown in the figures by eliminating states 16 and 17. Thus, the trellis is exemplary only.

[0039] The invention described in the above detailed description is not intended to be limited to the specific form set forth herein, but is intended to cover such alternatives, modifications and equivalents as can reasonably be included within the spirit and scope of the appended claims.

Claims

1. A Viterbi trellis characterized by being configurable to support an EPRML type channel (208) or an E2PRML type channel (208) and having identical connectivity for said EPRML and said E2PRML type channels.

2. A Viterbi trellis according to claim 1, wherein said EPRML type channel (208) is a channel having predetermined parity.

3. A Viterbi trellis according to claim 2, wherein said parity is 0 mod 2 parity.

4. A Viterbi trellis according to claim 3, said trellis being a sixteen state trellis wherein sixteen edges between states are eliminated every P clock cycles, where P is the block length of a received code.

5. A Viterbi trellis according to claim 1, wherein said E2PRML type channel comprises an ME2PRML type channel.

6. A Viterbi trellis according to claim 5, wherein said E2PRML type channel has four edges eliminated every k mod 9 clock cycles, where k is {1, 2, 4, 6, 8}.

7. A method for implementing a decoder, characterized by:

   providing a trellis, said trellis having predetermined connectivity; and
   configuring said trellis to support an EPRML type channel (208) or an E2PRML type channel (208).

8. A method according to claim 7, wherein said E2PRML type channel comprises an M2EPRML type channel and said E2PRML type channel comprises an ME2PRML type channel.

9. A method according to claim 7, wherein said providing said trellis comprises providing a sixteen-state trellis in which sixteen of thirty-two edges have same NRZ values in said EPRML type channel (208) and said E2PRML type channel (208).

10. A method according to claim 9, wherein said providing said trellis comprises providing a sixteen-state trellis in

which sixteen of thirty-two edges have same sample values in said EPRML type channel and said E2PRML type channel.

11. A method according to claim 10, wherein said configuring said EPRML type channel comprises eliminating sixteen edges every P clock cycles, where P is a block length of an associated code, every time a sample that corresponds to a last coordinate of a codeword reaches the trellis

12. A method according to claim 11, wherein said configuring said E2PRML type channel comprises eliminating four edges every k mod 9 clock cycles, where k is {1, 2, 4, 6, 8}.

13. A method, comprising:

defining a first state transition diagram;
defining a second state transition diagram isomorphic with said first state transition diagram; and
implementing an nth order channel filter and an (n-1)th order channel filter on said first and second state transition diagrams;
wherein a same underlying trellis is used to detect both said nth order channel filter and said (n-1)th order channel filter with parity.

14. A Viterbi decoder, comprising a trellis used to detect both an nth order channel filter and an (n-1)th order channel filter with parity, said nth order channel filter and said (n-1)th order channel filter being isomorphic.

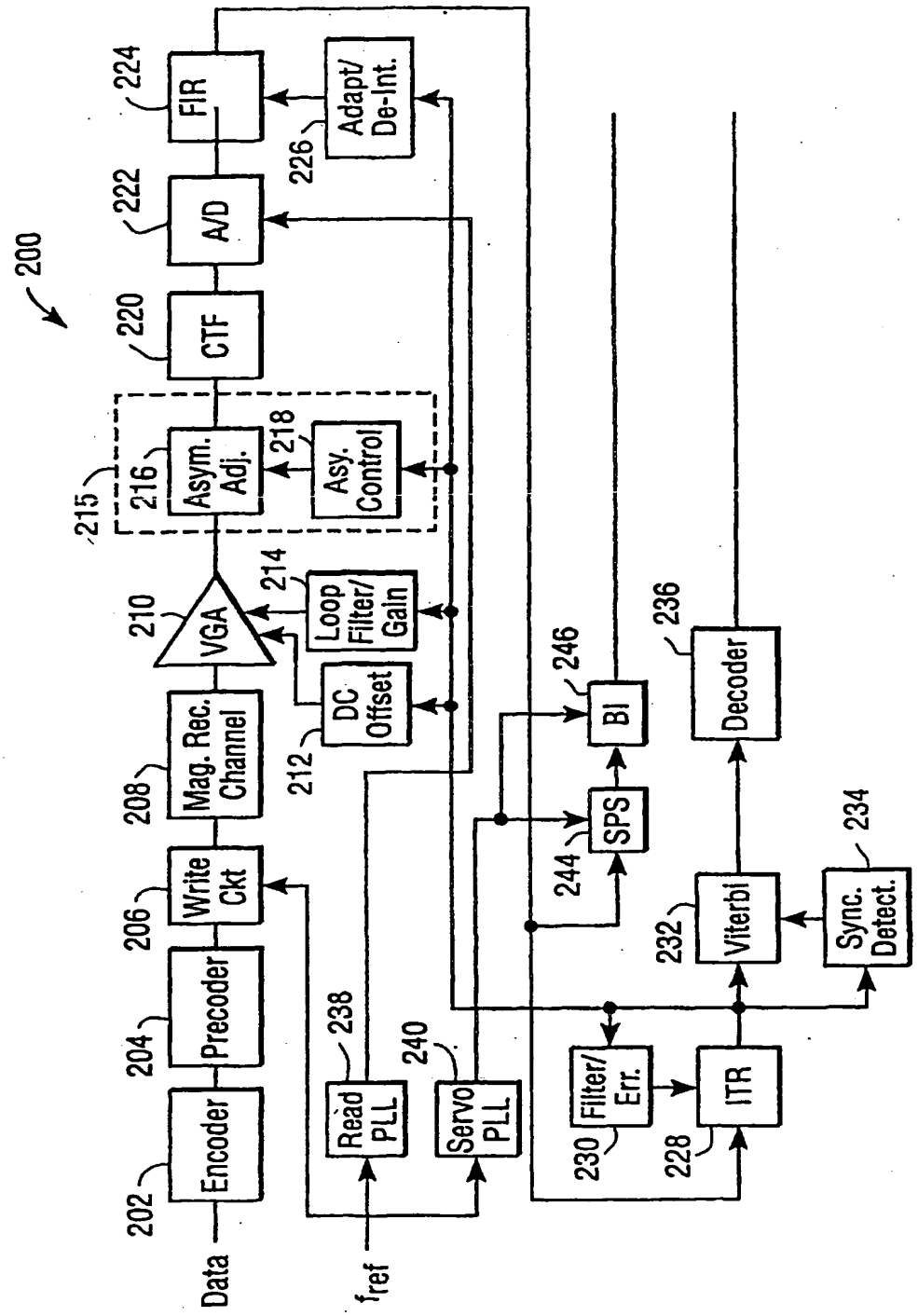
200
202
Encoder
204
Precoder
206
Write Ckt
208
Mag. Rec. Channel
210
VGA
212
DC Offset
214
Loop Filter/ Gain
215
216
Asym. Adj.
218
Asy. Control
220
CTF
222
A/D
224
FIR
226
Adapt/ De-Int.
228
ITR
230
Filter/ Err.
232
Viterbi
234
Sync. Detect.
236
Decoder
238
Read PLL
240
Servo PLL
244
SPS
246
BI
Data
f_{ref}

FIG. 1

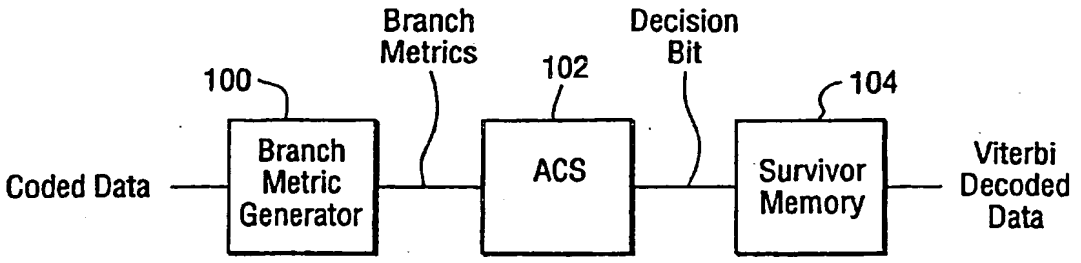
Branch
Metrics
Decision
Bit
100
102
104
Coded Data
Branch
Metric
Generator
ACS
Survivor
Memory
Viterbi
Decoded
Data

FIG. 2

T1 & T2 Trellises have identical connectivity
SOLID ARROW MEANS NRZ=1
DOTTED ARROW MEANS NRZ=0

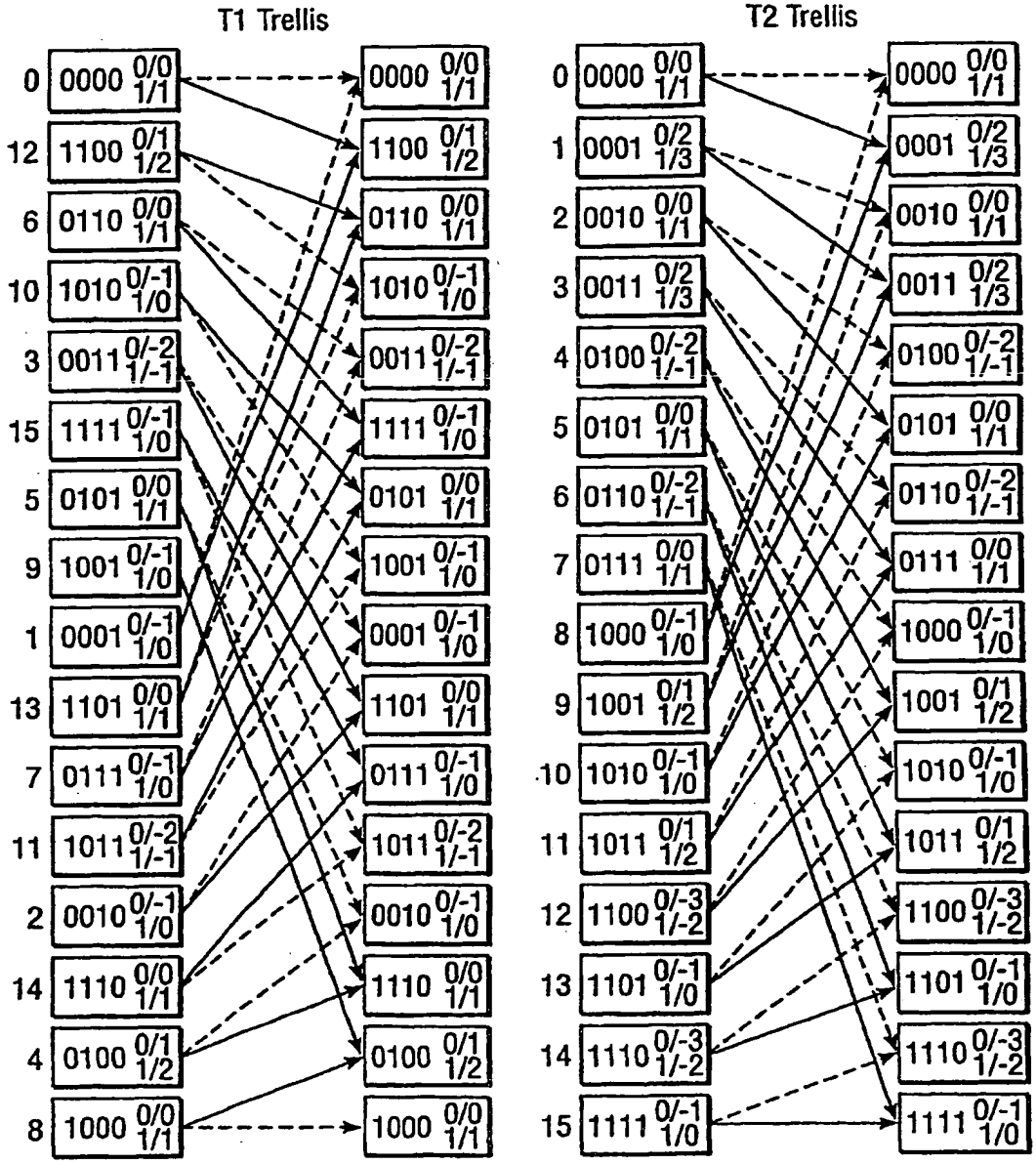

FIG. 3

Common NRZ values on T1 & T2 Trellises
16 out of 32
(Outgoing from even states in T2)
SOLID ARROW MEANS NRZ=1
DOTTED ARROW MEANS NRZ=0

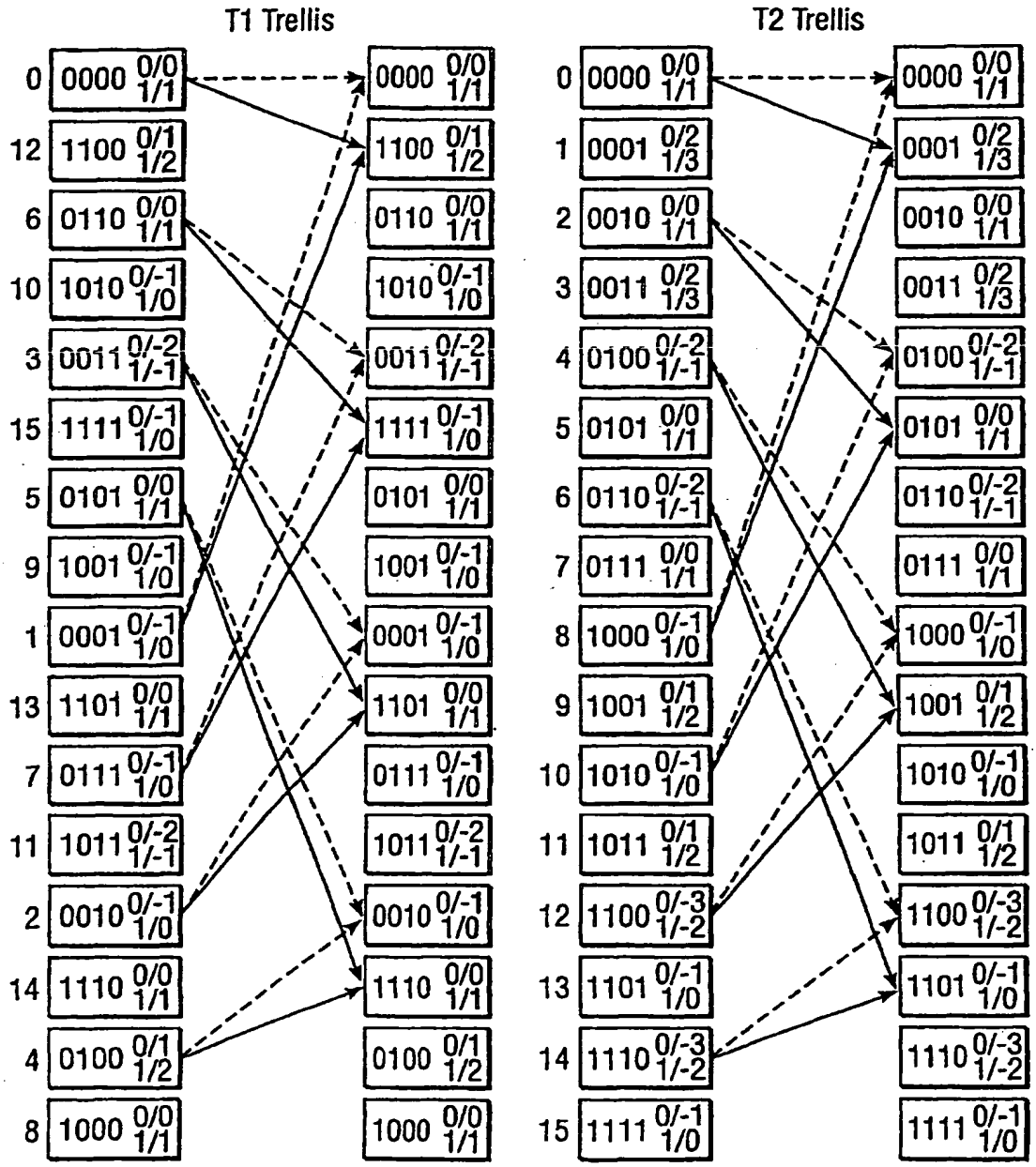

FIG. 4

Common Sample values on T1 & T2 Trellises
16 out of 32
SOLID ARROW MEANS NRZ=1
DOTTED ARROW MEANS NRZ=0

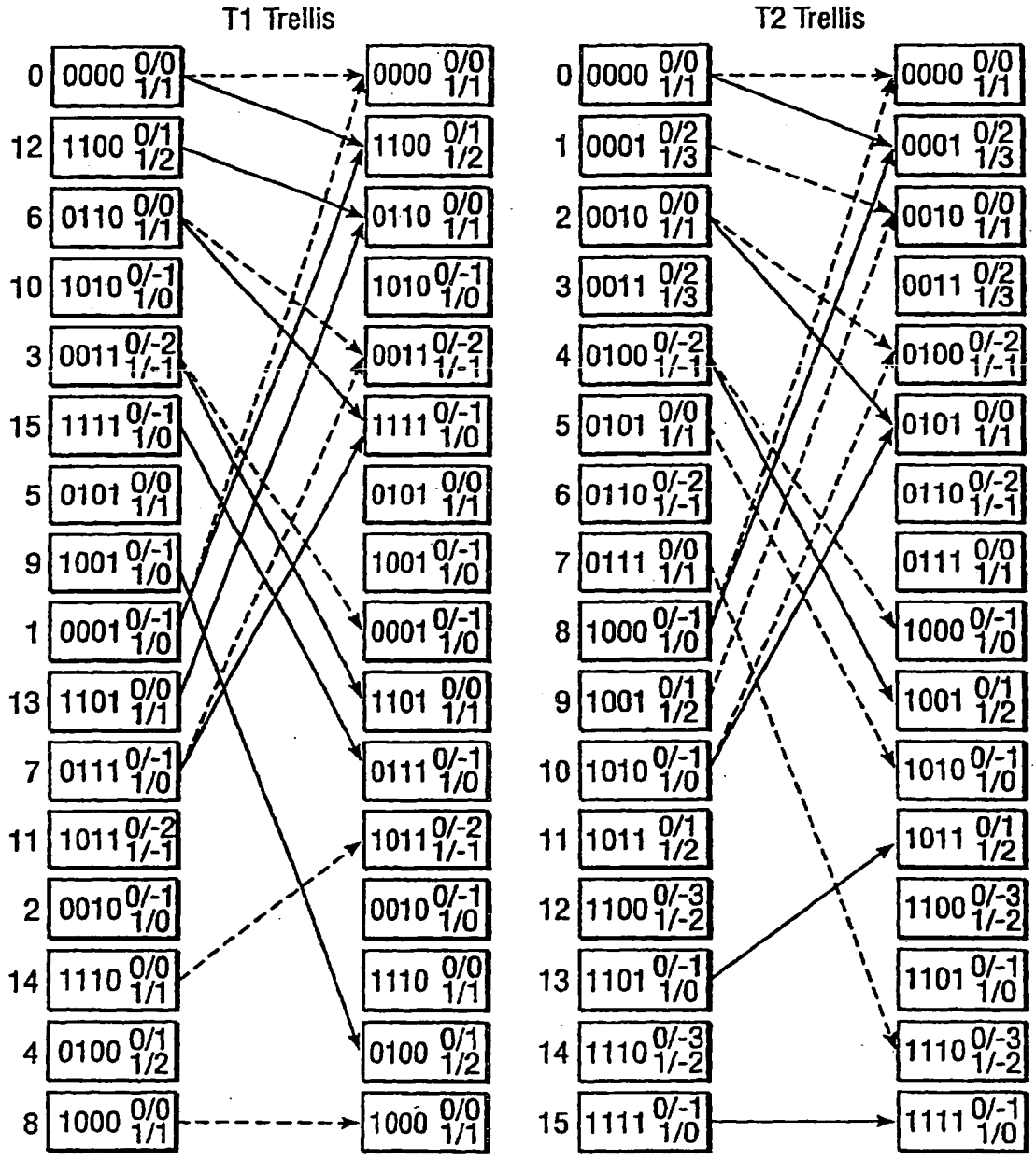

FIG. 5

EPRML 0 mod2 & ME2PRML Trellises based on
T1 & T2 Trellises
SOLID ARROW MEANS NRZ=1
DOTTED ARROW MEANS NRZ=0

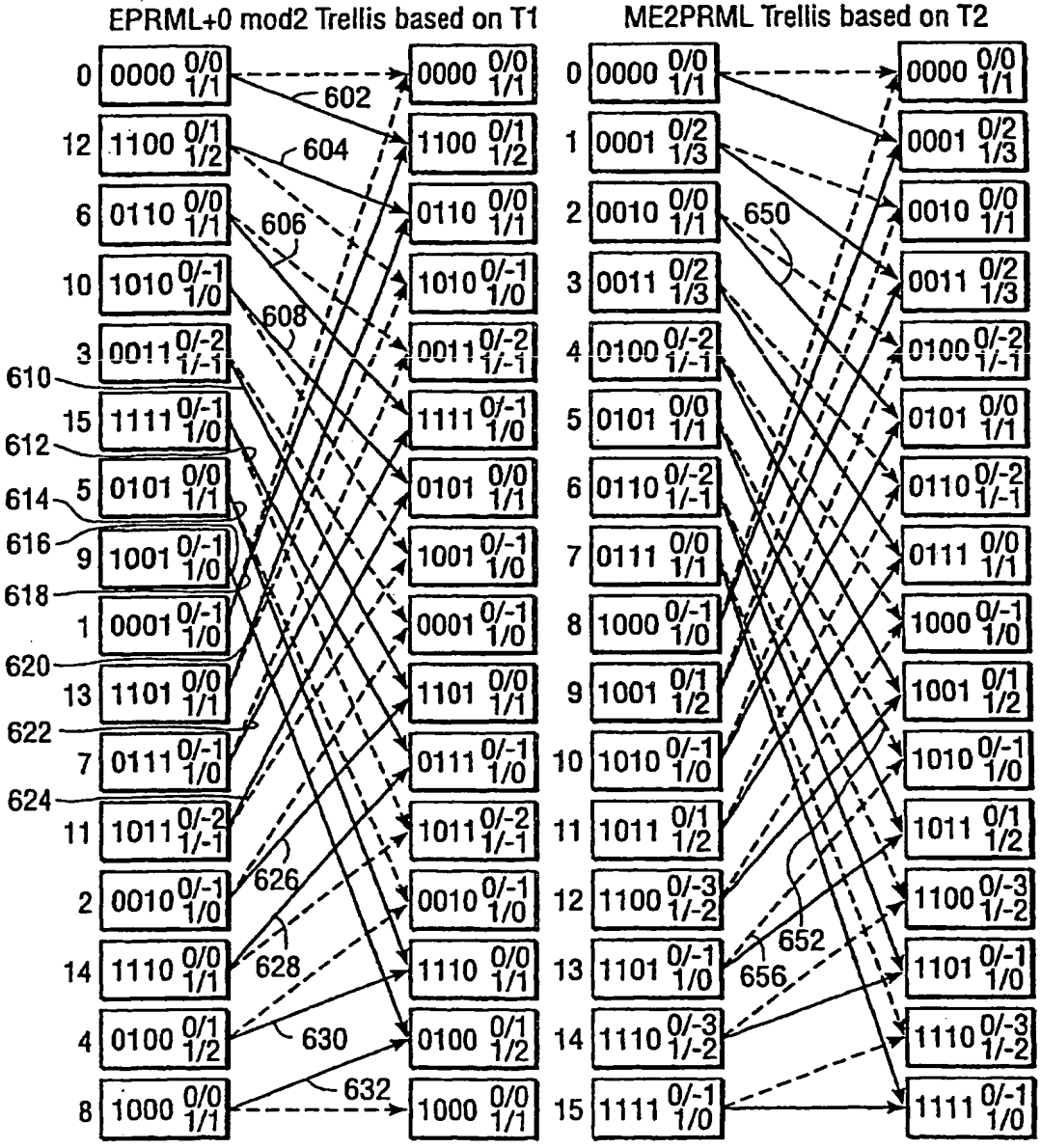

FIG. 6

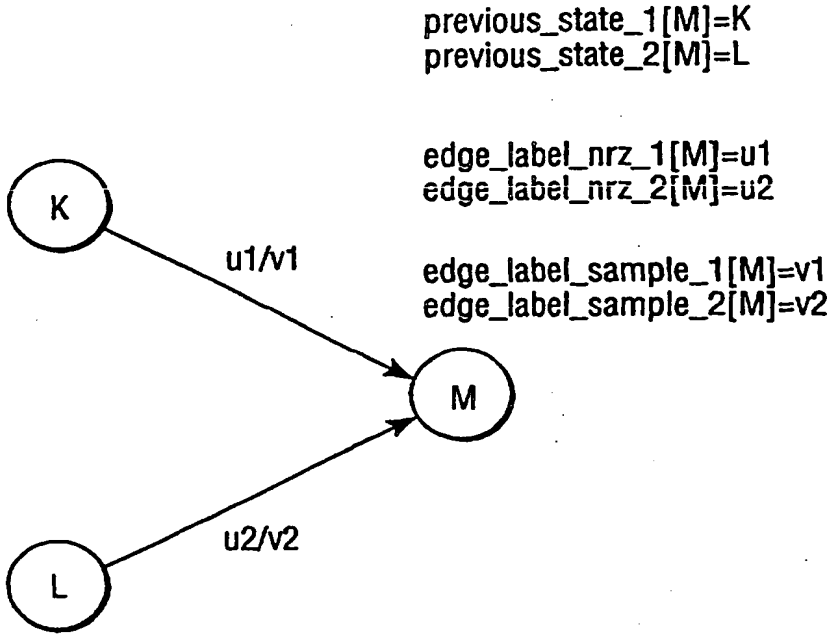
previous_state_1[M]=K
previous_state_2[M]=L
edge_label_nrz_1[M]=u1
edge_label_nrz_2[M]=u2
edge_label_sample_1[M]=v1
edge_label_sample_2[M]=v2
K
u1/v1
M
u2/v2
L

FIG. 7